線形符号

情報数理学特別講義I 第2回

2025-11-04

本日の内容

- きちんとした（数学的な）符号の定義と論理回路が少々
- 線形符号
- ハミング符号

符号の定義をきちんとする

("きちんと"してないかもしれないので不明点は都度質問してください.)

定義：符号

$GF(2)$ 上の長さ n の符号 $\mathcal{C}$ とは，$GF(2)^n$ の部分集合である.
そして符号 $\mathcal{C}$ の要素 $\underline{c} \in \mathcal{C}$ を符号語(codeword)と呼ぶ.

ただし，$GF(2)$は2元体（サイズ2のガロア体）で，次のスライドで定義を示す.

※ガロア体は有限体のこと．ガロア体については後ほど説明します.

2元体($GF(2)$)

0と1の2つの元を持ち，加算(Addition)と乗算(Multiplication)がそれぞれで下記のように定義される体を2元体 $GF(2)$ と呼ぶ.

Table 1: Addition

$+$	0	1
0	0	1
1	1	0

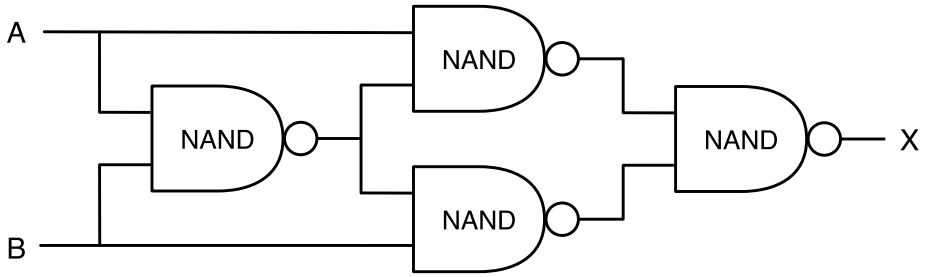
Table 2: Multiplication

$\cdot$	0	1
0	0	0
1	0	1

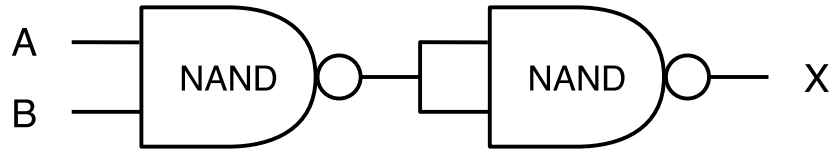
$GF(2)$の加算と乗算は，それぞれ論理演算における排他的論理和(XOR)と論理積(AND)である.
電気回路上ではそれぞれXORゲート，ANDゲートと呼ばれる論理回路で実装される.

XORゲート，ANDゲート

XORとAND回路を2入力NANDゲートで実装した図を以下に示す.
XORゲートは2入力NANDゲート4つで実現でき，乗算であるANDゲートは2入力NANDゲート2つで実現できる.



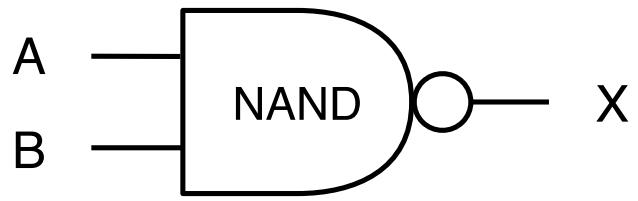
A	B	X
0	0	0
1	0	1
0	1	1
1	1	0



A	B	X
0	0	0
1	0	0
0	1	0
1	1	1

ユニバーサルゲートとしてのNANDゲート

半導体で論理回路を実装する際の最小構成単位は2入力NANDゲートである.
下図に2入力NANDゲートの回路図と真理値表を示す.



A	B	X
0	0	1
1	0	1
0	1	1
1	1	0

NANDゲートはユニバーサルゲートと呼ばれ，その組み合わせにより任意の論理回路を実現することができる．論理回路と半導体チップ面積は比例関係にあるため，半導体チップに実装した際のチップ面積を2入力NANDゲートの数で推定することができる.

きちんとした符号の定義

定義：符号

$GF(2)$ 上の長さ n の符号 $\mathcal{C}$ とは，$GF(2)^n$ の部分集合である.
そして符号 $\mathcal{C}$ の要素 $\underline{c} \in \mathcal{C}$ を符号語(codeword)と呼ぶ.

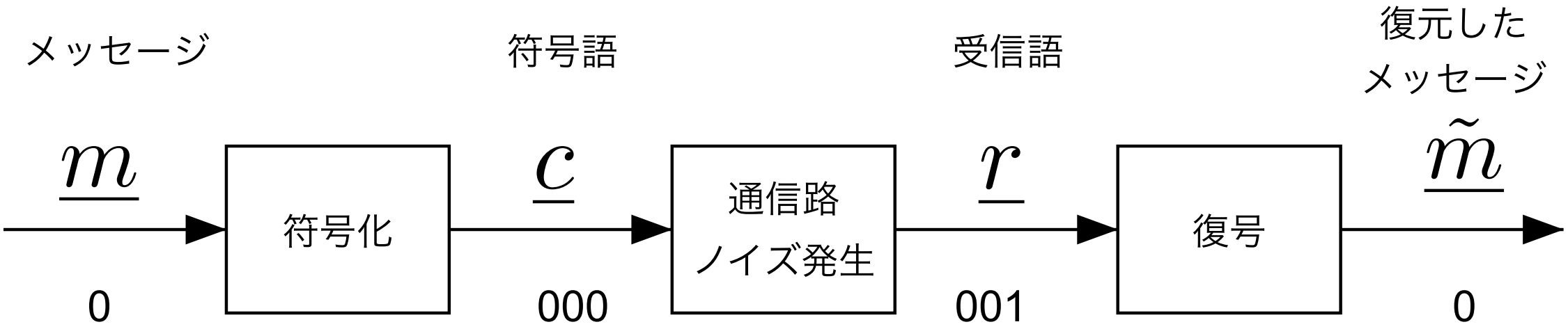
※この定義は$GF(2)$上での符号の定義だが，任意の体で符号は定義してもよい．本講義では$GF(2)$上の符号を議論する.

$GF(2)$は論理回路で実装できる

$GF(2)$の要素は0, 1のビットで，その演算は論理回路で実現できる.
任意の論理回路はユニバーサルゲートであるNANDゲートの組み合わせで表現できるため,
NANDゲート数によって回路規模を見積もることができる.

符号化と復号手順

符号は符号語（バイナリベクトル）の集合なので，符号化，復号の際には符号表が必要．



Message	Codeword
0	000
1	111

Distance

1

2

Codeword	Message
000	0
111	1

符号表を用いる符号化と復号は，メッセージサイズが大きくなると実現困難

復号時に符号表内の全ての符号語と受信語とのハミング距離を計算する必要がある.
符号表内の符号語数は 2^k なので，計算量は $O(2^k)$ となり，メッセージサイズが大きいと実現困難になる.

また，符号表のエントリ数も 2^k あるため，符号表をメモリ上に保持することも困難.

- 表を用いるのではなく，計算により符号化，復号できる符号が望ましい.

線形符号

定義：線形符号

$GF(2)$ 上の長さ n の線形符号 $\mathcal{C}$ とは,

$$\mathcal{C} = \{\underline{c} \mid H\underline{c}^{\mathrm{T}} = \underline{0}, \underline{c} \in GF(2)^n\}$$

を満たす$GF(2)^n$ の部分集合である．ただし，H は $n-k \times n$ の $GF(2)$ 上の行列（バイナリ行列）で，k は符号化するメッセージの長さである．なお，k は次元とも呼ぶ．また，$\underline{0}$ は要素が全て 0 の長さ $n-k$ のゼロベクトルである.

読み方：$\{\underline{c} \mid$ 条件 $\}$ で，条件を満たす $\underline{c}$ の集合を表す．ここでの条件は $H\underline{c}^{\mathrm{T}} = \underline{0}$ である.

線形符号は，$GF(2)$ 上の長さ n の線形空間上の，部分空間であると言っても良い.
※以後，線形符号に関する演算は線形空間を仮定する.

線形符号の例：単一パリティ検査符号

$n = 3, k = 2$ の単一パリティ検査符号は，次のパリティ検査行列との積が $\underline{0}$ となる長さ3のバイナリベクトルの集合と定義される．

$$H = (1 \quad 1 \quad 1).$$

練習問題： 長さ3の単一パリティ検査符号 $\mathcal{C} = \{(000), (101), (011), (110)\}$ なので，積が $\underline{0}$ となるか確かめよ．

線形符号の例：繰り返し符号

$n = 3, k = 1$ の繰り返し符号 $\mathcal{C} = \{(000), (111)\}$ も線形符号なので，パリティ検査行列が定義できる.

練習問題１： $n = 3, k = 1$ の繰り返し符号のパリティ検査行列 H を求めよ.
練習問題２： 符号語 $(000), (111)$ とパリティ検査行列 H の積が $\underline{0}$ となることを確かめよ.

パリティ検査とシンドロームベクトル

長さ n のベクトルとパリティ検査行列 H との積を計算することを**パリティ検査**と呼ぶ.
パリティ検査により得られる長さ $n-k$ のベクトルを**シンドロームベクトル**（単純にシンドロームと呼ぶこともある）と呼ぶ.

$$\underline{s}^T = H\underline{r}^T$$

問題：シンドロームが以下のとき，受信語 $\underline{r}$ はどのような状態か考えてみよう.

- $\underline{s}^T = \underline{0}$ のとき ...
- $\underline{s}^T \neq \underline{0}$ のとき ...

線形符号の有用な性質

- シンドロームを使った復号
- 生成行列を使った符号化

シンドロームを使った復号

線形符号はパリティ検査をすることで，誤りの有無を判定できる.
※ただし，符号の訂正能力内の誤り数に限る.

ここでは，シンドロームベクトルを利用した復号方法を述べる.

シンドロームを使った復号手順：

1. 受信語 $\underline{r}$ に対してパリティ検査を行い $\underline{s}^T = H\underline{r}^T$，シンドロームベクトル $\underline{s}$ を得る.
2. シンドロームベクトル $\underline{s}$ とエラーベクトル $\underline{e}$ の対応表 $\mathcal{T}$ を用いてエラーベクトルを求める，
   $\underline{e} = \mathcal{T}(\underline{s})$.
   ※エラーベクトルとは，誤りが発生した座標の要素が 1でそれ以外が 0 となるベクトル.
3. 受信語 $\underline{r}$ にエラーベクトル $\underline{e}$ を加えることで，復号語 $\underline{\tilde{c}} = \underline{r} + \underline{e}$ を得る.

$n = 3, k = 1$ の繰り返し符号のシンドローム復号

繰り返し符号のシンドローム表 $\mathcal{T}$ は以下の通り

$\underline{s}$	$\underline{e}$
(10)	(100)
(01)	(010)
(11)	(001)

ただし，Tは転置を表す.
いま，受信語 $\underline{r} = (101)$ とすると，シンドロームベクトル $\underline{s}$ は何になるか．さらにエラーベクトル $\underline{e}$，復号語 $\underline{\tilde{c}}$ を求めよ.

$n = 3, k = 1$ の繰り返し符号のシンドローム復号

$\underline{s}$	$\underline{e}$
(10)	(100)
(01)	(010)
(11)	(001)

受信語 $\underline{r} = (101)$ のとき,

$$\underline{s}^T = H\underline{r}^T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = (01)^T$$

したがって，シンドローム表 $\mathcal{T}$ から $\underline{e} = (010)$ が得られる.
そして，復号語 $\underline{\tilde{c}} = \underline{r} + \underline{e} = (101) + (010) = (111)$ が得られる.

問：線形符号はなぜ，符号語 $\underline{c}$ に依らず，シンドロームベクトル $\underline{s}$ だけで復号できるのか？

$\underline{0} = H\underline{c}$ より明らか

$$\underline{s}^T = H\underline{r}^T = H(\underline{c} + \underline{e})^T = H\underline{c}^T + H\underline{e}^T = \underline{0} + H\underline{e}^T = H\underline{e}^T$$

以上より，シンドロームベクトル $\underline{s}$ は，符号語 $\underline{c}$ に依らず，エラーベクトル $\underline{e}$ にのみ依存することがわかる.

シンドローム表の大きさは 2^{n-k} であるので，符号化率が 1/2 より大きい符号では，シンドローム表を用いた復号のほうがテーブルサイズが小さい $2^{n-k} < 2^k$. また，符号表を用いる復号では符号語ごとのハミング距離が最小になる符号語を探索しなければならなかったのに対し，シンドローム表を用いる復号では，シンドロームベクトルが一致するエラーベクトルを探索すればよいだけなので，復号処理が高速になる.

生成行列を使った符号化

線形符号の符号化は，次の演算により実施できる

$$\underline{c} = \underline{m}G$$

ただし，G は $GF(2)$ 上の k 行 n 列の行列（バイナリ行列）で，生成行列と呼ぶ．$\underline{m}$ は長さ k のメッセージである．

例：長さ $k = 3$ の繰り返し符号の生成行列による符号化
生成行列 $G = (111)$ より，

- $0(111) \rightarrow (000)$
- $1(111) \rightarrow (111)$

生成行列の作成方法については後ほど紹介する．

線形符号のまとめ

- 線形符号の定義：

$$\mathcal{C} = \{\underline{c} \mid H\underline{c} = \underline{0}, \underline{c} \in GF(2)^n\}$$

ただし，Hは$n-k \times n$の$GF(2)$上の行列（バイナリ行列）.

- シンドロームベクトル$\underline{s}$，は長さnのベクトルとパリティ検査行列Hとの積で得られる長さ$n-k$のバイナリベクトル.
  - $\underline{s} = 0$ならば誤りなし．さもなければ誤りあり.
- シンドローム表による符号語に依らず復号できる.
- メッセージと生成行列の積により符号化が可能.

1ビット訂正可能な繰り返し符号の符号化率は 1/3.

2/3 は冗長データということ.

もっと効率よく（高い符号化率で）符号化できないか？

ハミング符号

長さ $n = 2^m - 1$ のハミング符号は，長さ m の相異なる非ゼロベクトル全てを列に持つパリティ検査行列 H で定義される線形符号.
つまり,

$$\mathcal{C} = \{\underline{c} \mid H\underline{c} = \underline{0}, \underline{c} \in GF(2)^n\}$$

ただし，Hは $n-k \times n$ の $GF(2)$ 上の行列（バイナリ行列）であり

$$H = (b_m(1), b_m(2), \cdots, b_m(n))$$

と定義される．ただし，$b_m(i)$ は整数 i の 2 進法表現の長さ m のバイナリベクトルである．例えば，$b_3(6) = (011)^T$.

長さ $n = 7$ のハミング符号

長さ $n = 7$ のハミング符号はパリティ検査行列

$$
\begin{aligned}
H &= (b_3(1), b_3(2), b_3(3), b_3(4), b_3(5), b_3(6), b_3(7)) \\
&= \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}
\end{aligned}
$$

で定義される.

ハミング符号の符号化率

ハミング符号のパリティビット（冗長ビット）数は m ビットである.
符号長 $n = 2^m - 1$ で，メッセージの長さは $k = n - m$ であることから，符号化率は

$$R = \frac{k}{n} = \frac{n-m}{n} = 1 - \frac{m}{2^m - 1}$$

である.
m が大きくなるほど，符号化率が1に漸近していくことがわかる（効率がよい）.
例えば $m = 8$ で $R = 0.9686$ となる.

符号の最小ハミング距離と最小重み

1ビット訂正可能であることは，最小ハミング距離が3であることを示せば良い.

- t ビット訂正可能な符号が少なくとも達成しなければいけない最小距離は $2t+1$.

符号語数が多いと距離を調べるのが大変. . .
しかし，線形符号の最小ハミング距離は，符号語の最小重みを調べることでわかる.

- 重み：ベクトルに含まれる非ゼロ要素の数．バイナリベクトルならば 1 の数.

符号の最小ハミング距離と最小重み

$GF(2)$ 上のベクトル $\underline{u}, \underline{v}$ 間のハミング距離 $d_H(\underline{u}, \underline{v})$ は ベクトル $\underline{w} = \underline{u} + \underline{v}$ の重み $wt(\underline{w})$ と一致する.

- ベクトル $\underline{w}$ の i 番目の要素 w_i が 1 のとき，つまり $w_i = 1$ は $u_i \neq v_i$ の場合だから.

線形符号では符号語の和も符号語である.

- $\underline{c}_3 = \underline{c}_1 + \underline{c}_2$ のとき，$H\underline{c}_3 = H(\underline{c}_1 + \underline{c}_2) = H\underline{c}_1 + H\underline{c}_2 = \underline{0} + \underline{0} = \underline{0}$ となる.

つまり，符号語間の相違を表すベクトル自体が符号語となるため，符号語の重みを調べることと符号語間のハミング距離を調べることは同値である.

よって，**線形符号の最小ハミング距離は，符号語の最小重みを調べることで求まる**.

ハミング符号の最小距離（最小重み）は3である

最小重みが2より大きいことの証明：

重み2の符号語を持たない $\Leftrightarrow$ パリティ検査行列が重複した列ベクトルを持たない.

ハミング符号のパリティ検査行列は**相異なる**非零ベクトル全てを列に持つので，重み2の符号語を持たない．つまりハミング符号の最小重みは2より大きい

重み3の符号語を持つことの証明：

a 行目にのみ 1 を持つ列ベクトルの列番号を i_a，b 行目にのみ 1 を持つ列ベクトルの列番号を i_b，a 行目と b 行目にのみ 1 を持つ列ベクトルの列番号を i_{a+b} とする時，ハミング符号は i_a, i_b, i_{a+b} に 1 を持つ重み3の符号語を持つ.

以上より，ハミング符号の最小距離（最小重み）は 3 である.

ハミング符号の生成行列

線形符号の生成行列は以下の2通りで生成することができる.

1. 線形独立な長さ n の符号語を k 個選び，それらを行ベクトルとして並べた行列を生成行列とする.
2. パリティ検査行列 H を $\tilde{H} = (PI_{n-k})$ の形に行基本変形し，$G = (I_k P^T)$ とする.

$$HG^T = \tilde{H}G^T = (PI_{n-k}) \begin{pmatrix} I_k \\ P \end{pmatrix} = (P + P) = \mathbf{0}$$

ただし I_k は $k \times k$ の単位行列.
生成行列の作り方はどちらの方法でもよいが，2.の方法の場合，符号語に含まれる最初の k ビットにメッセージがそのまま現れる．このように符号化することを組織符号化と呼び，このように符号化を行う符号を組織符号(systematic code)と呼ぶ.

長さ $n = 7$ のハミング符号の生成行列

長さ $n = 7$ のハミング符号はパリティ検査行列

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

を $\tilde{H} = (PI_{n-k})$ の形に行基本変形($r(3) \leftarrow r(2) + r(3), r(1) \leftrightarrow r(3), r(3) \leftarrow r(3) + r(1)$, $r(2) \leftarrow r(2) + r(3)$)すると,

$$\tilde{H} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

長さ $n = 7$ のハミング符号の生成行列（続き）

$$P = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}.$$

よって，生成行列 $G = (I_k P^T)$ より

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

練習問題(3分)： $HG^T = \mathbf{0}$ になることを確かめよ.

ハミング符号のまとめ

- 長さ m の相異なる非零ベクトル全てを列に持つパリティ検査行列 H で定義される符号
- 符号化率は $R = 1 - \frac{m}{2^m - 1}$
- 最小距離は3で1ビット訂正可能.
- ハミング符号はハミング限界を達成する完全符号である.

演習

課題1：$n = 5, k = 1$の繰り返し符号のパリティ検査行列 H を示し，対応するシンドローム表を作成せよ.
課題2：課題1で示したパリティ検査行列 H と作成したシンドローム表を用いて，受信語 $r = (10101)$ の復号を行え．ただし，復号語だけでなく復号手順も示すこと.
課題3：ハミング符号が完全符号であることを証明せよ．なお，完全符号とは前回の講義で説明したハミング限界を等号で満たす符号のことである.